CLAIMS

What is claimed is:

1. A method of executing an interpretive language in a system having a processing component with native software processes, and a memory component, comprising the steps of:

coupling a hardware component with the processing component and the memory component;

employing the hardware component to assist processing of the interpretive language; and

permitting the system to execute the native software processes of the processing component.

2. The method of claim 1, comprising the step of generating an instruction jump address at the hardware component.

3. The method of claim 2, comprising the step of maintaining a current address for the interpretive language instruction stream at the hardware component.

4. The method of claim 3, comprising of the step of automatically incrementing the current address for the interpretive language instruction stream at the hardware component in response to an address sent from the processing component and received at the hardware component corresponding to a fixed instruction fetch address stored at the hardware component.

5. The method of claim 2, in which the generating of the instruction jump address is in response to the hardware component determining that an address received from the processing component corresponds to a fixed instruction fetch address stored at the hardware component.

6. The method of claim 5, comprising the step of transmitting a current interpretive language address from the hardware component to the memory component upon the hardware component determining that the address received from the processing component corresponds to the fixed instruction fetch address.

7. The method of claim 6, comprising the step of transmitting the instruction jump address from the hardware component to the processing component.

8. The method of claim 7, comprising the steps of fetching data associated with the current interpretive language address from the memory component,

transmitting the data from the memory component to the hardware component wherein the data comprises at least a current interpretive instruction,

shifting the current interpretive instruction by a predetermined number of bits at the hardware component to establish a shifted address, and

adding the shifted address to a base address stored in the hardware component to calculate the instruction jump address.

9. The method of claim 1, comprising the step of comparing at the hardware component an address received from the processing component to a fixed instruction fetch address stored at a decoding component of the hardware component.

10. The method of claim 9, comprising the steps of determining at the decoding component of the hardware component that the address received from the processing component does not correspond to the fixed instruction fetch address, and

sending the address received from the processing component to the memory component.

11. The method of claim 9, comprising the step of determining at the decoding component of the hardware component that the address received from the processing component corresponds to the fixed instruction fetch address stored at the decoding component.

12. The method of claim 10, comprising the step of maintaining a current interpretive language address in an interpreter language program counter of the hardware component.

13. The method of claim 12, comprising the steps of loading the interpreter language program counter with a starting address of an interpretive language instruction stream, and

automatically incrementing the interpreter language program counter to the next address of the interpretive language instruction stream upon the address read from the processing component corresponding to the fixed instruction fetch address.

14. The method of claim 12, comprising the step of transmitting a select signal from the decoding component to an address multiplexer component of the hardware component.

15. The method of claim 14, comprising the step of transmitting the current interpretive language address from the interpreter language program counter of the hardware

component to the memory component via the address multiplexer component based on the select signal received at the address multiplexer component from the decoding component.

16. The method of claim 15, comprising the step of fetching data from a location at the memory component which is associated with the current interpretive language address.

17. The method of claim 16, comprising the step of transmitting the data from the memory component to an instruction jump address generator component of the hardware component,

wherein the data comprises at least a current interpretive instruction.

18. The method of claim 17, comprising the steps of shifting the current interpretive instruction by a predetermined number of bits at the instruction jump address generator component to establish a shifted address, and

adding the shifted address to a base address stored in the instruction jump address generator component to calculate an instruction jump address.

19. The method of claim 18, comprising the step of transmitting the instruction jump address from the instruction jump address generator component of the hardware component to a data multiplexer component of the hardware component.

20. The method of claim 19, comprising the step of transmitting a data multiplexer select signal from the decoding component to the data multiplexer component of the hardware component based on the address received at the decoding component from the processing component corresponding to the fixed instruction fetch address.

21. The method of claim 20, comprising the step of transmitting the instruction jump address from the data multiplexer component of the hardware component to the processing component based on the data multiplexer select signal from the decoding component.

22. The method of claim 1, comprising the step of fetching from the memory component at least one operand of an interpretive language instruction stream.

23. The method of claim 22, comprising the step of maintaining operand addresses at the hardware component.

24. The method of claim 22, in which the operand addresses are maintained at an interpreter language program counter of the hardware component.

25. The method of claim 22 comprising the steps of fetching data which comprises at least one operand from the memory component, and

storing the data comprising at least one operand at the hardware component.

26. The method of claim 25, comprising the step of sending the data from the memory component via a data bus capable of carrying more than one operand to the hardware component.

27. The method of claim 26, in which the data comprises at least one of : (a) an 8-bit operand, (b) a 16-bit operand, and (c) a 24-bit operand.

28. The method of claim 25, comprising the steps of determining at the hardware component an operand size requested to be read by the processing component, and

transmitting the operand requested by the processing component from the hardware component to the processing component.

29. The method of claim 28, comprising the step of determining at the hardware component if an address received from the processing component corresponds to a fixed operand fetch address stored at the hardware component.

30. The method of claim 22, comprising the step of comparing at a decoding component of the hardware component an address received from the processing component to a number of fixed operand fetch address stored at the decoding component.

31. The method of claim 30, comprising the step of determining at the decoding component that the address is equal to a fixed operand fetch address.

32. The method of claim 30, comprising the step of transmitting an operand address from an interpreter language program counter of the hardware component to the memory component.

33. The method of claim 31, comprising the step of transmitting a select signal from the decoding component to an address multiplexer component of the hardware component.

34. The method of claim 33, comprising the step of fetching data associated with the operand address from the memory component.

35. The method of claim 34, comprising the step of transmitting the data from the memory component via a data bus to an operand storing component of the hardware component, and

wherein the data comprises at least one operand of the interpretive language instruction stream.

36. The method of claim 35, in which the data bus is a 32-bit data bus such that four bytes of 8-bit data are sent from the memory component to the operand storing component.

37. The method of claim 35, comprising the step of determining at the decoding component an operand size requested to be read by the processing component, and

transmitting the operand of the size requested by the processing component from the operand storing component to the processing component.

38. The method of claim 37, comprising the steps of transmitting a data multiplexer select signal to a data multiplexer component of the hardware component in response to the decoding component determining that the address received from the processing component corresponds to the fixed operand fetch address, and

transmitting the operand to the processing component via the data multiplexer component based on the data multiplexer select signal received from the decoder component.

39. The method of claim 38, comprising the steps of

storing multiple bytes at the operand storing component, and

shifting the multiple operand bytes at the operand storing component into an order required by the processing component.

40. The method of claim 1, comprising the step of storing a thread switch jump address at the hardware component.

41. The method of claim 40, comprising the step of transmitting the thread switch jump address from the hardware component to processing component.

42. The method of claim 41, comprising the steps of determining at the hardware component if an address received from the processing component is equal to a fixed instruction fetch address, and

incrementing a counter component of the hardware component in response to the determination that the address received from the processing component corresponds to the fixed instruction fetch address.

43. The method of claim 42, comprising the steps of counting a total number of addresses received from the processing component which are determined to correspond to the fixed instruction fetch address, and

determining that the total number counted has reached a predetermined number.

44. The method of claim 43, in which the step of determining if an address received from the processing component is equal to the fixed instruction fetch address is performed at a decoding component of the hardware component.

45. The method of claim 44, comprising the step of transmitting the thread switch jump address from the counter component to a data multiplexer component of the hardware component.

46. The method of claim 45, comprising the steps of transmitting a data multiplexer select signal from the decoder component to the data multiplexer component in response to the total number counted at the counter component reaching the predetermined number, and

transmitting the thread switch jump address from the counter component to the processing component via the data multiplexer component based on the data multiplexer select signal.

47. A system which executes an interpretive language having a processing component with native software processes and a memory component comprising:

a hardware component coupled with the processing component and the memory component to assist processing of the interpretive language with the system able to execute the native software processes of the processing component.

48. The system of claim 47, further comprising a decoding component of the hardware component that stores a fixed instruction fetch address, said decoding component compares an address received from the processing component with the fixed instruction fetch address.

49. The system of claim 48, in which the decoding component determines if the address received from the processing component corresponds to the fixed instruction fetch address.

50. The system of claim 49 in which the address received from the processing component is sent to the memory component via the hardware component in response to the

decoding component determining that the address received does not correspond to the fixed instruction fetch address.

51. The system of claim 49 further comprising an interpreter language program counter of the hardware component which maintains a current interpretive language address.

52. The system of claim 51 in which the interpreter language program counter is loaded with a starting address of an interpretive language instruction stream, said interpreter language program counter is automatically incremented to the next address of the interpretive language instruction stream upon the address read from the processing component corresponding to the fixed instruction fetch address.

53. The system of claim 51 further comprising an address multiplexer component of the hardware component which receives a select signal transmitted from the decoding component.

54. The system of claim 53 in which the interpreter language program counter transmits the current interpretive language address to the memory component via the address multiplexer component based on the select signal received at the address multiplexer component from the decoding component.

55. The system of claim 54 in which data is fetched from a location at the memory component which is associated with the current interpretive language address.

56. The system of claim 55 further comprising an instruction jump address generator component of the hardware component which receives the data transmitted from the memory component, wherein the data comprises at least a current interpretive instruction.

57. The system of claim 56 in which the instruction jump address generator component shifts the current interpretive instruction by a predetermined number of bits to establish a shifted address, said instruction jump address generator component stores a base address which is added to the shifted address to calculate an instruction jump address.

58. The system of claim 57 further comprising a data multiplexer component of the hardware component which receives the instruction jump address transmitted from the instruction jump address generator component.

59. The system of claim 58 in which the decoding component transmits a data multiplexer select signal to the data multiplexer component based on the address received at the decoding component from the processing component corresponding to the fixed instruction fetch address.

60. The system of claim 59 in which the instruction jump address is transmitted from the data multiplexer component to the processing component based on the data multiplexer select signal from the decoding component.

61. The system of claim 47 in which at least one operand of an interpretive language instruction stream is fetched from the memory component.

62. The system of claim 61 further comprising an interpreter language program counter of the hardware component for maintaining operand addresses.

63. The system of claim 61 in which the hardware component stores data which comprises at least one operand which is fetched from the memory component.

64. The system of claim 63 further comprising a data bus coupled with the memory component and the hardware component which is capable of carrying more than one operand of data sent from the memory component to the hardware component.

65. The system of claim 64 in which the data comprises at least one of: (a) an 8-bit operand, (b) a 16-bit operand, and (c) a 24-bit operand.

66. The system of claim 64 in which the hardware component determines an operand size requested to be read by the processing component and the hardware component transmits the operand of the size requested by the processing component to the processing component.

67. The system of claim 66 in which the hardware component determines if an address received from the processing component corresponds to one of a number of fixed operand fetch address stored at the hardware component.

68. The system of claim 61 further comprising a decoding component of the hardware component which compares an address received from the processing component to the fixed operand fetch addresses stored at the decoding component.

69. The system of claim 68 in which the decoding component determines that the address corresponds to one of the fixed operand fetch addresses.

70. The system of 68 further comprising an interpreter language program counter of the hardware component which transmits an operand address to the memory component.

71. The system of claim 69 in which the decoding component transmits a select signal to an address multiplexer of the hardware component.

72. The system of claim 71 in which data associated with the operand address is fetched from the memory component.

73. The system of claim 72 further comprising an operand storing component of the hardware component which receives data transmitted from the memory component via a data bus to the operand storing component wherein the data comprises at least one operand of the interpretive language instruction stream.

74. The system of claim 73 in which the data bus is a 32-bit data bus such that four bytes of 8-bit data are sent from the memory component to the operand storing component.

75. The system of claim 73 in which the decoding component determines an operand size requested to be read by the processing component and the operand storing component transmits the operand of the size requested to the processing component.

76. The system of claim 75 further comprising a data multiplexer component of the hardware component which receives a data multiplexer select signal in response to the decoding component determining that the address received from the processing component is equal to a fixed operand fetch address, said operand is transmitted to the processing component via the data multiplexer component based on the data multiplexer select signal received from the decoder component.

77. The system of claim 76 in which the operand storing component stores multiple operands and the operand storing component shifts the multiple operands into an order required by the processing component.

78. The system of claim 47 further comprising a thread switch jump address stored at the hardware component.

79. The system of claim 78 in which the hardware component transmits the thread switch jump address to the processing component.

80. The system of claim 79 further comprising a counter component of the hardware component which is incremented in response to a determination at the hardware component that an address received from the processing component corresponds to a fixed instruction fetch address.

81. The system of claim 80 in which the counter component counts a total number of addresses received from the processing component which are determined by a decoding component of the hardware component to correspond to the fixed instruction fetch address and the counter component determines that the total number counted has reached a predetermined number.

82. The system of claim 81 further comprising a data multiplexer component of the hardware component which receives the thread switch jump address transmitted from the counter component.

83. The system of claim 82 in which a data multiplexer select signal is transmitted from the decoder component to the data multiplexer component in response to the total number counted at the counter component reaching a predetermined number and the thread switch jump address is transmitted from the counter component to the processing component via the data multiplexer component based on the data multiplexer select signal.